(19)国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 118092855 A
(43)申请公布日 2024.05.28

(21)申请号 202410142246.6

(22)申请日 2024.02.01

(71)申请人 复旦大学
地址 200433 上海市杨浦区邯郸路220号

(72)发明人 解玉凤 易明铮 万文章 林昱凯 钟海凡 马梓钊 陈董觎

(74)专利代理机构 上海德昭知识产权代理有限公司 31204
专利代理师 程宗德

(51)Int.Cl.
G06F 7/498(2006.01)

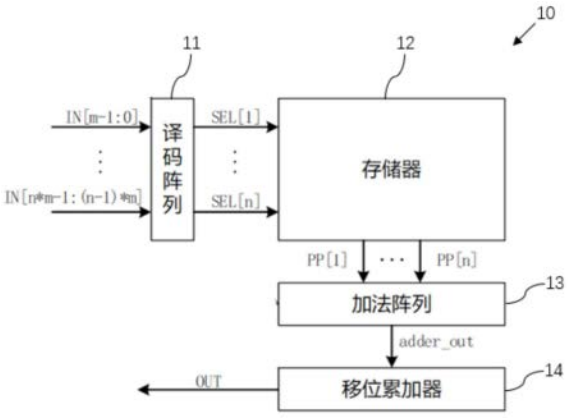
权利要求书2页 说明书8页 附图4页

(54)发明名称

支持浮点数尾数乘法的存算一体乘法器及乘法运算方法

(57)摘要

本发明提供一种支持浮点数尾数乘法的存算一体乘法器,其至少包括译码阵列、存储器以及加法阵列,利用该乘法器进行乘法运算时,被乘数以部分积的形式配置在存储器内,乘数以特定位宽为单位划分成片段,以片段为单位串行地加载到译码阵列输入端,直到完成一个完整乘数的输入,译码阵列根据输入的乘数片段选通存储器中相应存储器行,存储器输出相应部分积,加法阵列基于部分积进行错位相加并输出乘法结果,与传统存算结构相比,该乘法器可大大减少部分积累加的次数,从而降低动态功耗,提高运算速度。



CN 118092855 A

1.一种支持浮点数尾数乘法的存算一体乘法器,其特征在于,包括:

译码阵列,包含至少一个地址译码器,用于对输入的乘数的多个乘数片段进行地址译码,得到译码结果;

存储器,其输入端与所述译码阵列的输出端连接,用于存储被乘数的多个部分积并根据所述译码结果输出相应的所述部分积;

加法阵列,其输入端与所述存储器的输出端连接,用于对所述部分积进行错位相加,输出加法结果;以及

移位累加器,其输入端与所述加法阵列的输出端连接,用于对所述加法结果进行移位累加,得到乘法结果。

2.根据权利要求1所述的支持浮点数尾数乘法的存算一体乘法器,其特征在于:

其中,所述地址译码器的数量为n个,每个所述地址译码器的位宽为m位,用于输入m位的所述乘数片段,

所述部分积为所述被乘数与所有可能的m位比特数的遍历乘法结果。

3.根据权利要求1所述的支持浮点数尾数乘法的存算一体乘法器,其特征在于:

其中,所述存储器包括:

多个存储单元;

多条位线,用于读或写对应的所述存储单元,所述存储单元按行分组地连接到对应的所述位线上;以及

位线选通器,所述多条位线分别连接至所述位线选通器的读写端,所述译码阵列的输出端连接至所述位线选通器的输入端,所述位线选通器用于根据所述译码结果选通相应的所述位线,从而选通对应的所述存储单元的行。

4.一种利用权利要求1-3中任意一项所述的支持浮点数尾数乘法的存算一体乘法器进行乘法运算的方法,其特征在于,包括:

将乘数以预定位宽为单位划分成多个乘数片段,并将多个所述乘数片段串行地加载到所述译码阵列的输入端;

所述译码阵列对所述乘数片段进行地址译码得到译码结果,并将所述译码结果向所述存储器输出;

所述存储器根据所述译码结果,将其存储的相应的被乘数的部分积向所述加法阵列输出;

所述加法阵列基于所述部分积进行错位相加得到加法结果,并将所述加法结果向所述移位累加器输出;

所述移位累加器对所述加法结果进行移位累加,得到乘法结果。

5.一种支持浮点数尾数乘法的高位宽存算一体乘法器,其特征在于,包括:

多个支持浮点数尾数乘法的存算一体乘法器;以及

错位相加器,多个支持浮点数尾数乘法的存算一体乘法器的输出端分别连接至所述错位相加器的输入端,所述错位相加器用于对多个所述支持浮点数尾数乘法的存算一体乘法器的乘法结果进行错位相加,得到高位宽乘法结果,

其中,所述支持浮点数尾数乘法的存算一体乘法器为权利要求1-3中任意一项所述的支持浮点数尾数乘法的存算一体乘法器。

6.根据权利要求5所述的支持浮点数尾数乘法的高位宽存算一体乘法器,其特征在于:

其中,多个所述支持浮点数尾数乘法的存算一体乘法器的所述译码阵列用于输入相同的所述乘数片段,

多个所述支持浮点数尾数乘法的存算一体乘法器的多个所述存储器分别用于存储所述部分积的不同的权重位段,

所述错位相加器根据对应的所述权重位段将多个所述乘法结果进行移位相加,从而得到所述高位宽乘法结果。

7.一种利用如权利要求5或6所述的支持浮点数尾数乘法的高位宽存算一体乘法器进行乘法运算的方法,其特征在于,包括:

将乘数以预定位宽为单位划分成多个乘数片段,并将所述多个乘数片段串行地加载到多个所述支持浮点数尾数乘法的存算一体乘法器的所述译码阵列的输入端;

多个所述译码阵列分别对所述乘数片段进行地址译码得到译码结果,并将所述译码结果向对应的所述存储器输出;

多个所述存储器分别根据所述译码结果,将其存储的相应的被乘数的部分积向对应的所述加法阵列输出;

多个所述加法阵列分别基于相应的所述部分积进行错位相加得到加法结果,并将所述加法结果向对应的所述移位累加器输出;

在所述乘数的所有比特位输入完毕后,多个所述移位累加器分别对相应的多个加法结果进行移位累加,得到乘法结果,并将所述乘法结果向所述错位相加器进行输出;

所述错位相加器将多个所述乘法结果错位相加,得到所述高位宽乘法结果。

支持浮点数尾数乘法的存算一体乘法器及乘法运算方法

技术领域

[0001] 本发明属于集成电路设计技术领域,具体涉及一种支持浮点数尾数乘法的存算一体乘法器以及乘法运算方法。

背景技术

[0002] 近几年来,随着大数据和人工智能领域技术的快速发展,数据密集型和计算密集型的应用场景越来越多。而且这些应用场景往往有着高访存性、大规模并行性、多重复性的特点。数据量大、计算量多对集成电路的高速低功耗提出了更加严峻的考验。现有的冯诺依曼架构中存储和计算分离的机制,使得消耗在数据传输上的能量和时间,在消耗的总能量和总时间中占比十分巨大,不能满足运算发展的需求。此外,在科学计算和高性能计算领域,例如自动驾驶、航空航天、力学计算领域,核心运算类型往往是浮点运算。传统处理器中的浮点运算单元通常消耗的资源、功耗都很大。当执行高并行的浮点乘加运算时,则需要更多的浮点单元,消耗巨大的资源和功耗。因此,亟需新型的运算架构来克服这一瓶颈。

[0003] 存算一体技术是在存储器里或者附近直接进行运算,能解决计算密集型应用中访问存储器所带来的数据搬运开销,能提高运算的速度和能效,是一种有潜力的运算架构。现有技术中存在一些存算一体乘法器,但这些存算一体乘法器仍有其不足之处。

[0004] 例如,文献【Ping-Chun Wu1,Jian-Wei Su,et al.A 22nm 832Kb Hybrid-Domain Floating-Point SRAMIn-Memory-Compute Macro with16.2-70.2TFLOPS/W for High-Accuracy AI-Edge Devices[C]//2023IEEE International Solid-State Circuits Conference(ISSCC).IEEE,2023,65:1-3.】提出了基于SRAM的数模混合型存算一体结构,用于神经网络推理。它的存算结构支持BFP16型浮点乘加运算,对于浮点权重的尾数采用三角阵列存储,按列加权排布的方法。这种存算一体结构虽然实现了较大的数据输入并行度,但不足在于三角阵列的存储方式导致SRAM的利用率不高,如果将此架构拓展到FP32模式,依然采用三角阵列的方式来排放尾数,那么将会形成更多的SRAM冗余阵列,面积效率很低。

[0005] 文献【Lee J,Kim J,Jo W,et al.A 13.7TFLOPS/W floating-point DNN processor using heterogeneous computing architecture with exponent-computing-in-memory[C]//2021Symposium on VLSI Circuits.IEEE,2021:1-2.】提出了基于SRAM的“异构”型存算一体结构,同样是用于神经网络推理。它的存算结构支持FP32型浮点乘加运算,对于浮点权重的指数采用SRAM阵列配合逻辑门来处理,浮点权重的尾数采用SRAM阵列外围的数字电路来处理。这种存算一体结构虽然可以实现浮点运算,但不足在于仅优化了指数运算,没有优化功耗、速度开销大的尾数运算,因此无法有效的发挥存算一体的低功耗优势。

[0006] 专利CN2021108325336中提出了一种基于DRAM和查找表的存算一体浮点乘加结构。在移位对齐方面,利用循环移位产生所有指数对齐可能需要的结果,然后遍历查找表搜索目标操作数。在浮点尾数乘法方面,遍历n比特数据片段和源尾数乘法,而后将部分积直接存在DRAM里作为查找表,查找累加。这种存算一体浮点乘加结构可将部分积累加次数从m

降低到m/n,但不足之处在于,第一它的硬件不支持可重构,也就是说只能支持一种数据位宽,不能重构成支持多种位宽的运算。第二它的输入并行度只能取决于单个译码器的输入地址位宽,并行度难以很好的提高。

[0007] 如上所述,现有技术中的存算一体结构存在扩展性差、效率提升有限的问题,因此,需要一种更易扩展的支持浮点数尾数乘法的存算一体结构。

发明内容

[0008] 本发明是为解决上述问题而进行的,目的在于提供一种更易扩展且支持浮点数尾数乘法的存算一体乘法器以及利用该乘法器进行乘法运算的方法,本发明采用了如下技术方案:

[0009] 本发明提供了一种支持浮点数尾数乘法的存算一体乘法器,其特征在于,包括:译码阵列,包含至少一个地址译码器,用于对输入的乘数的多个乘数片段进行地址译码,得到译码结果;存储器,其输入端与所述译码阵列的输出端连接,用于存储被乘数的多个部分积并根据所述译码结果输出相应的所述部分积;加法阵列,其输入端与所述存储器的输出端连接,用于对所述部分积进行错位相加,输出加法结果;以及移位累加器,其输入端与所述加法阵列的输出端连接,用于对所述加法结果进行移位累加,得到乘法结果。

[0010] 本发明提供的支持浮点数尾数乘法的存算一体乘法器,还可以具有这样的技术特征,其中,所述地址译码器的数量为n个,每个所述地址译码器的位宽为m位,用于输入m位的所述乘数片段,所述部分积为所述被乘数与所有可能的m位比特数的遍历乘法结果。

[0011] 本发明提供的支持浮点数尾数乘法的存算一体乘法器,还可以具有这样的技术特征,其中,所述存储器包括:多个存储单元;多条位线,用于读或写对应的所述存储单元,所述存储单元按行分组地连接到对应的所述位线上;以及位线选通器,所述多条位线分别连接至所述位线选通器的读写端,所述译码阵列的输出端连接至所述位线选通器的输入端,所述位线选通器用于根据所述译码结果选通相应的所述位线,从而选通对应的所述存储单元的行。

[0012] 本发明提供了一种利用上述支持浮点数尾数乘法的存算一体乘法器进行乘法运算的方法,其特征在于,包括以下步骤:将乘数以预定位宽为单位划分成多个乘数片段,并将多个所述乘数片段串行地加载到所述译码阵列的输入端;所述译码阵列对所述乘数片段进行地址译码得到译码结果,并将所述译码结果向所述存储器输出;所述存储器根据所述译码结果,将其存储的相应的被乘数的部分积向所述加法阵列输出;所述加法阵列基于所述部分积进行错位相加得到加法结果,并将所述加法结果向所述移位累加器输出;所述移位累加器对所述加法结果进行移位累加,得到乘法结果。

[0013] 本发明提供了一种支持浮点数尾数乘法的高位宽存算一体乘法器,其特征在于,包括:多个上述的支持浮点数尾数乘法的存算一体乘法器;以及错位相加器,多个支持浮点数尾数乘法的存算一体乘法器的输出端分别连接至所述错位相加器的输入端,所述错位相加器用于对多个所述支持浮点数尾数乘法的存算一体乘法器的乘法结果进行错位相加,得到高位宽乘法结果。

[0014] 本发明提供的支持浮点数尾数乘法的高位宽存算一体乘法器,还可以具有这样的技术特征,其中,多个所述支持浮点数尾数乘法的存算一体乘法器的所述译码阵列用于输

入相同的所述乘数片段,多个所述支持浮点数尾数乘法的存算一体乘法器的多个所述存储器分别用于存储所述部分积的不同的权重位段,所述错位相加器根据对应的所述权重位段将多个所述乘法结果进行移位相加,从而得到所述高位宽乘法结果。

[0015] 本发明提供了一种利用上述支持浮点数尾数乘法的高位宽存算一体乘法器进行乘法运算的方法,其特征在于,包括以下步骤:将乘数以预定位宽为单位划分成多个乘数片段,并将所述多个乘数片段串行地加载到多个所述支持浮点数尾数乘法的存算一体乘法器的所述译码阵列的输入端;多个所述译码阵列分别对所述乘数片段进行地址译码得到译码结果,并将所述译码结果向对应的所述存储器输出;多个所述存储器分别根据所述译码结果,将其存储的相应的被乘数的部分积向对应的所述加法阵列输出;多个所述加法阵列分别基于相应的所述部分积进行错位相加得到加法结果,并将所述加法结果向对应的所述移位累加器输出;在所述乘数的所有比特位输入完毕后,多个所述移位累加器分别对相应的多个加法结果进行移位累加,得到乘法结果,并将所述乘法结果向所述错位相加器进行输出;所述错位相加器将多个所述乘法结果错位相加,得到所述高位宽乘法结果。

[0016] 发明作用与效果

[0017] 根据本发明的支持浮点数尾数乘法的存算一体乘法器以及相应的乘法运算方法,与现有技术的存算结构相比,具有以下优势:

[0018] 在输入位宽方面,本发明的存算一体乘法器支持多地址输入,总输入位宽为地址数乘以地址位宽。地址位宽越高、相应配置的存储器的查找阵列越大,数据处理的速度就越快,因此,本发明的存算一体乘法器能大大提高数据处理速度。

[0019] 在功耗以及运算速度方面,在硬件层面上,乘法运算的本质是部分积的错位相加,传统存算结构需要先根据动态的输入,算出单比特部分积,而后再送到后续的错位相加电路来完成累加,与之相比,本发明的存算一体乘法器则是直接读取存储器的存储阵列中存储的多比特遍历部分积编码而后进行压缩相加、错位相加,因此大大减少了部分积累加的次数,有利于降低动态功耗,提高运算速度。因此,无论是在功耗还是速度方面,本发明的存算一体乘法器都显著优于传统预存储权重的存算架构以及背景技术中所提到的存算一体结构。

附图说明

[0020] 图1是本发明实施例中存算一体乘法器的结构示意图;

[0021] 图2是本发明实施例中存储器的结构示意图;

[0022] 图3是本发明实施例中利用存算一体乘法器进行乘法运算的流程图;

[0023] 图4是本发明实施例中高位宽存算一体乘法器的结构示意图;

[0024] 图5是本发明实施例中利用高位宽存算一体乘法器进行乘法运算的流程图。

[0025] 附图标记:

[0026] 存算一体乘法器10;译码阵列11;存储器12;加法阵列13;移位累加器14;高位宽存算一体乘法器100;错位相加器20。

具体实施方式

[0027] 为了使本发明实现的技术手段、创作特征、达成目的与功效易于明白了解,以下结

合实施例及附图对本发明的支持浮点数尾数乘法的存算一体乘法器以及乘法运算方法作具体阐述。

[0028] <实施例一>

[0029] 本实施例提供一种支持浮点数尾数乘法的存算一体乘法器,用于进行浮点数的尾数的乘法运算。为更好说明该乘法器的作用,以下将先对浮点数乘法的过程进行简要说明。

[0030] 已知浮点数A和浮点数B,分别以S_A、E_A、M_A代指浮点数A的符号位、指数位、尾数位(包含隐藏位“1”),分别以S_B、E_B、M_B代指浮点数B的符号位、指数位、尾数位(包含隐藏位“1”),以E_S代表浮点的指数偏移量,则对浮点数A和浮点数B进行乘法运算(即求解Y=A×B)的流程包括以下步骤:

[0031] 步骤S1,将浮点数A的符号位S_A与浮点数B的符号位S_B进行异或运算,即:

$$S_Y = S_A \oplus S_B$$。

[0032] 步骤S2,将浮点数A的指数位E_A与浮点数B的指数位E_B相加,即:$E_Y=E_A+E_B-E_S$。

[0033] 步骤S3,将浮点数A的尾数位M_A与浮点数B的尾数位M_B相乘,即:$M_Y=M_A\times M_B$。

[0034] 步骤S4,将上述步骤的计算结果归一化成浮点格式,从而得到浮点数A和浮点数B的乘法运算结果。

[0035] 上述四个步骤示出了浮点数的乘法运算中所涉及到的所有运算类型,其中,尾数乘法步骤S3是整个浮点数的乘法运算中耗费功耗时间面积成本最大的,本实施例的存算一体乘法器正是针对步骤S3而做出的设计优化。

[0036] 图1是本实施例中支持浮点数尾数乘法的存算一体乘法器的结构示意图。

[0037] 如图1所示,支持浮点数尾数乘法的存算一体乘法器10包括译码阵列11、存储器12、加法阵列13以及移位累加器14。

[0038] 译码阵列11包含一个或多个地址译码器,为便于叙述,记作包含n个地址译码器,每个地址译码器的地址位宽为m位,n和m可以是任意非零自然数。相应地,译码阵列11具有n位输入端以及n位输出端。其输入端用于输入n个乘数片段,输出端连接至存储器12的行字线。译码阵列11用于对输入的乘数的片段进行地址译码,并将相应的译码结果输出至存储器12。

[0039] 存储器12具有多位输入端(包括上述行字线)以及多位输出端,其输出端连接至加法阵列13的输入端。存储器12用于存储被乘数的部分积。

[0040] 图2是本实施例中存储器的结构示意图。

[0041] 如图2所示,本实施例的存储器12包括多个存储单元121、位线选通器122、位线BL[1]至BL[n],每条位线上连接k个存储单元121,形成k行n列存储阵列。位线均连接至位线选通器122,位线选通器122的输入线SEL连接至译码阵列11的输出端,位线选通器122根据来自译码阵列11的译码结果选通相应的位线。

[0042] 加法阵列13具有多位输入端以及多位输出端,其输出端连接至移位累加器14的输入端。加法阵列13用于将输入的多个加数相加得到一个相加结果的数。

[0043] 移位累加器14具有多位输入端以及多位输出端,其输出端用于输出乘法结果。移位累加器14用于对依次输入的多个数进行加权后再进行累加。

[0044] 在进行乘法运算时,被乘数以部分积的形式配置在存储器12内,部分积具体为被乘数的m bit部分积,即被乘数与所有可能的m bit数的遍历乘法结果;乘数则以特定位宽

为单位划分成多个乘数片段,以片段为单位串行地加载到译码阵列11的输入端,直到完成一个完整乘数的输入。

[0045] 译码阵列11对输入的n个乘数片段进行地址译码,并将相应的译码结果输出至存储器12,以选通n个存储器行。

[0046] 存储器12将n个选通的存储器行中的数作为n个加数输出至加法阵列13。

[0047] 加法阵列13将n个加数进行错位相加,输出加法结果。根据乘数片段的数量,加法阵列13将按照时钟周期依次输出多个加法结果。

[0048] 移位累加器14将来自加法阵列13的多个和数进行加权后再累加,输出乘法结果数。

[0049] 以下将示例性地具体示出利用上述存算一体乘法器10进行乘法运算的流程。

[0050] 在该示例性的乘法运算中,乘数为IN,被乘数为W,IN和W都是8bit原码整型,上述n=2,m=2。译码阵列11包含两个2-4地址译码器。存储器12具有4行10列存储单元121,并且支持两个查找地址同时查找。

[0051] 在开始乘法运算之前,需要先计算被乘数W的部分积,具体地,计算2bit索引与被乘数W的乘法的结果,即将2'd0、2'd1、2'd2、2'd3分别与W进行相乘,得到W的2bit部分积PP[0]、PP[1]、PP[2]、PP[3],其中,每个部分积都是10bit。

[0052] 然后,将计算出的部分积以查找表的形式存储在存储器12内,具体地,将上述四个部分积以索引值为行地址按行存储在存储器12内。之后,就可以输入乘数IN,开始进行乘法运算。

[0053] 图3是本实施例中利用存算一体乘法器进行乘法运算的流程图。

[0054] 如图3所示,该乘法运算可分解为以下步骤:

[0055] 步骤S1-1,在第一个时钟周期,向译码阵列11输入乘数片段IN[1:0]、IN[3:2],译码阵列11向存储器12输出相应的译码结果。

[0056] 步骤S1-2,存储器12基于译码结果,查找出被乘数W的两个部分积PP[IN[1:0]]、PP[IN[3:2]],并向加法阵列13输出两个部分积。

[0057] 步骤S1-3,加法阵列13基于两个部分积进行加法运算,得到加法结果$adder_out_{clk1}$=PP[IN[1:0]]+2^2*PP[IN[3:2]],并向移位累加器14输出加法结果。

[0058] 步骤S1-4,在第二个时钟周期,向译码阵列11输入乘数片段IN[5:4]、IN[7:6],译码阵列11向存储器12输出相应的译码结果。

[0059] 步骤S1-5,存储器12基于译码结果,查找出被乘数W的两个部分积PP[IN[5:4]]、PP[IN[7:6]],并向加法阵列13输出两个部分积。

[0060] 步骤S1-6,加法阵列13基于两个部分积进行错位相加,得到加法结果$adder_out_{clk2}$=PP[IN[5:4]]+2^2*PP[IN[7:6],并向移位累加器14输出加法结果。

[0061] 步骤S1-7,移位累加器14对两个时钟周期的加法结果进行加权后再进行累加,得到乘法结果

[0062] OUT=$adder_out_{clk1}$+2^4*$adder_out_{clk2}$=IN*W,并将乘法结果进行输出。

[0063] 实施例一作用与效果

[0064] 根据本实施例提供的支持浮点数尾数乘法的存算一体乘法器10,与现有技术的存算结构相比,具有以下优势:

[0065] 在输入位宽方面,本实施例的存算一体乘法器10支持多地址输入,总输入位宽为地址数乘以地址位宽,即:m*n。地址位宽越高、相应配置的存储器的查找阵列越大,数据处理的速度就越快,例如对于具有8bit尾数的高精度浮点数BFP16,可以将译码阵列11配置成m=4,n=2,将存储器12配置成具有16行12列存储单元,这样一来,一节拍即可输入一个BFP16浮点数,这是基于预存储权重原理的传统存算架构所不能达到的。

[0066] 在功耗以及运算速度方面,在硬件层面上,乘法运算的本质是部分积的错位相加,传统存算结构需要先根据动态的输入,算出单比特部分积,而后再送到后续的错位相加电路来完成累加,与之相比,本实施例的存算一体乘法器10则是直接读取存储器12的存储阵列中存储的多比特遍历部分积编码而后进行压缩相加、错位相加,因此大大减少了部分积累加的次数,有利于降低动态功耗,提高运算速度。因此,无论是在功耗还是速度方面,本实施例的存算一体乘法器10都显著优于传统预存储权重的存算架构以及背景技术中所提到的存算一体结构。

[0067] <实施例二>

[0068] 基于实施例一提供的支持浮点数尾数乘法的存算一体乘法器,本实施例提供一种支持浮点数尾数乘法的高位宽存算一体乘法器。

[0069] 图4是本实施例中高位宽存算一体乘法器的结构示意图。

[0070] 如图4所示,高位宽存算一体乘法器100包含多个实施例一的存算一体乘法器10以及错位相加器20。为便于叙述,将存算一体乘法器10记作q个,q是任意大于1的自然数,将q个存算一体乘法器10分别记作INST[1]至INST[q],将其存储器12分别记作MEM[1]至MEM[q]。

[0071] q个存算一体乘法器10的输入来自相同的乘数片段,其输出端依次连接至错位相加器20的多个输入端,存算一体乘法器10将其乘法结果输出至错位相加器20。

[0072] 错位相加器20将来自不同存算一体乘法器10的乘法结果按不同的权重段移位相加,输出一个高位宽乘法结果。

[0073] 以下将示例性地具体示出利用上述高位宽存算一体乘法器100进行乘法运算的流程。

[0074] 在该示例性的乘法运算中,乘数IN和被乘数W都是16bit原码整型,n=2,m=2,q=2,也即高位宽存算一体乘法器100包含两个存算一体乘法器10。

[0075] 同样地,在进行乘法运算前,需要先计算被乘数W的部分积,具体地,计算任意2bit索引与W的乘法结果,即将2'd0、2'd1、2'd2、2'd3分别与W相乘,得到W的2bit部分积PP[0]、PP[1]、PP[2]、PP[3],其中,每个部分积都是18bit。

[0076] 然后,将计算出的部分积以查找表的形式存储在存储器12内,具体地,将上述四个部分积的高9bit(即PP[x][17:9])以索引值为行地址按行存储在存储器MEM[1]内,并将上述四个部分积的低9bit(即PP[x][8:0])以索引值为行地址按行存储在存储器MEM[2]内。之后,就可以输入乘数IN,开始进行乘法运算。

[0077] 图5是本实施例中利用高位宽存算一体乘法器进行乘法运算的流程图。

[0078] 如图5所示,该乘法运算可分解为以下步骤:

[0079] 步骤S2-1,在第一个时钟周期,向两个译码阵列11分别输入乘数片段IN[1:0]、IN[3:2],两个译码阵列11分别向对应的存储器MEM[1]、MEM[2]输出相应的译码结果。

[0080] 步骤S2-2,存储器MEM[1]基于相应的译码结果,查找出被乘数W的两个部分积INST[1]:$PP_{MEM[1]}$[IN[1:0]]、$PP_{MEM[1]}$[IN[3:2]]并向相应的加法阵列13输出;

[0081] 存储器MEM[2]基于相应的译码结果,查找出被乘数W的两个部分积INST[2]:$PP_{MEM[2]}$[IN[1:0]]、$PP_{MEM[2]}$[IN[3:2]],并向相应的加法阵列13输出。

[0082] 步骤S2-3,存算一体乘法器INST[1]的加法阵列13分别基于对应的两个部分积进行错位相加运算,得到加法结果$adder_out_{clk1}=PP_{MEM[1]}[IN[1:0]]+2^2*PP_{MEM[1]}[IN[3:2]]$,并向对应的移位累加器14输出;

[0083] 存算一体乘法器INST[2]的加法阵列13分别基于对应的两个部分积进行错位相加运算,得到加法结果$adder_out_{clk1}=PP_{MEM[2]}[IN[1:0]]+2^2*PP_{MEM[2]}[IN[3:2]]$,并向对应的移位累加器14输出。

[0084] 步骤S2-4,在第二个时钟周期,向两个译码阵列11输入乘数片段IN[5:4]、IN[7:6],两个译码阵列11分别向对应的存储器MEM[1]、MEM[2]输出相应的译码结果。

[0085] 步骤S2-5,存储器MEM[1]基于相应的译码结果,查找出被乘数W的两个部分积INST[1]:$PP_{MEM[1]}$[IN[5:4]]、$PP_{MEM[1]}$[IN[7:6]]并向相应的加法阵列13输出;

[0086] 存储器MEM[2]基于相应的译码结果,查找出被乘数W的两个部分积INST[2]:$PP_{MEM[2]}$[IN[5:4]]、$PP_{MEM[2]}$[IN[7:6]],并向相应的加法阵列13输出。

[0087] 步骤S2-6,存算一体乘法器INST[1]的加法阵列13分别基于对应的两个部分积进行错位相加运算,得到加法结果$adder_out_{clk2}=PP_{MEM[1]}[IN[5:4]]+2^2*PP_{MEM[1]}[IN[7:6]]$,并向对应的移位累加器14输出;

[0088] 存算一体乘法器INST[2]的加法阵列13分别基于对应的两个部分积进行错位相加运算,得到加法结果$adder_out_{clk2}=PP_{MEM[2]}[IN[5:4]]+2^2*PP_{MEM[2]}[IN[7:6]]$,并向对应的移位累加器14输出。

[0089] 在步骤S2-6之后,省略了若干步骤,参照步骤S2-1至步骤S2-6,以此类推,直到将乘数IN的所有比特位都输入完毕,进入步骤S2-7。

[0090] 步骤S2-7,两个移位累加器14分别对多个时钟周期的相应的加法结果进行加权后再进行累加,得到乘法结果,并将乘法结果向错位相加器20进行输出。

[0091] 步骤S2-8,错位相加器20将多个乘法结果按不同权重段移位相加,得到高位宽乘法结果$OUT=shift_out[1]*2^9+shift_out[2]=IN*W$。

[0092] 实施例二作用与效果

[0093] 根据本实施例提供的支持浮点数尾数乘法的高位宽存算一体乘法器100,由于包含多个存算一体乘法器10以及错位相加器20,因此在实施例一的作用与效果的基础上,能够实现更高位宽的乘法运算,并且该结构采用模块化的设计,易于扩展,可方便地扩展至各种所需位宽。

[0094] 上述实施例仅用于举例说明本发明的具体实施方式,而本发明不限于上述实施例的描述范围,本行业的技术人员应该了解,本发明不受上述实施例的限制,上述实施例和说明书中描述的只是说明本发明的原理,在不脱离本发明精神和范围的前提下,本发明还会有各种变化和改进,这些变化和改进都落入要求保护的本发明范围内。本发明要求保护范围由所附的权利要求书及其等效物界定。

[0095] 例如,在上述实施例中,存算一体乘法器包括移位累加器,而对于位宽较少的乘数

来说,只需要查找一次存储器,然后通过加法阵列进行相加就可得出乘法结果,因此对于仅处理位宽较少的乘数的情况,存算一体乘法器也可以不具有移位累加器。

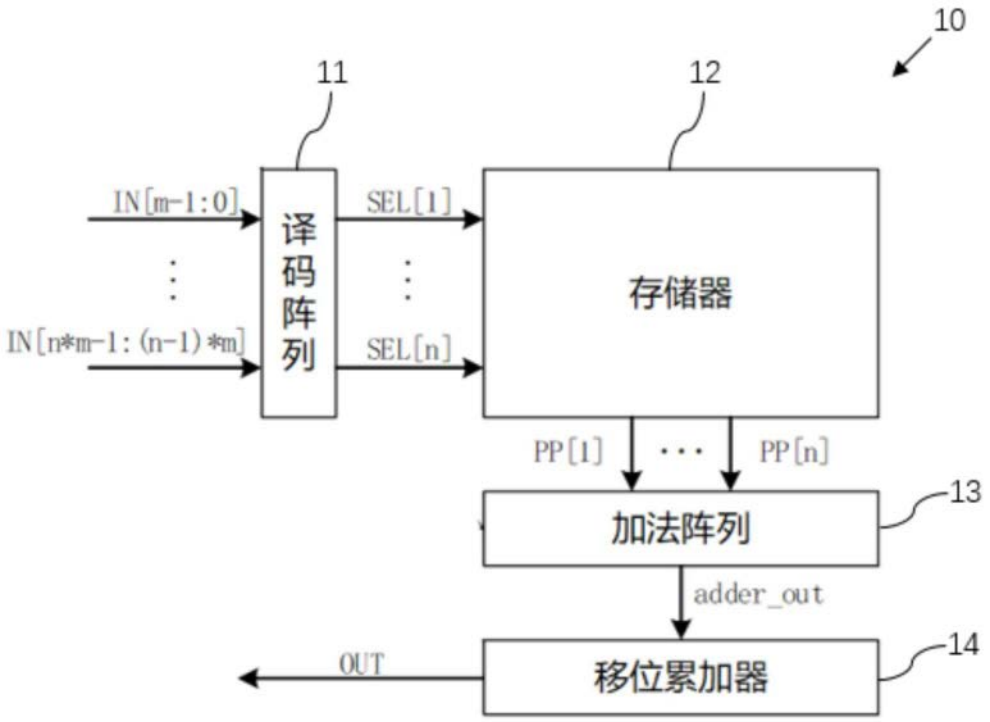

10
11
12
IN[m-1:0]
IN[n*m-1:(n-1)*m]
译码阵列
SEL[1]
SEL[n]
存储器
PP[1]
PP[n]
13
加法阵列
adder_out
OUT
14
移位累加器

图1

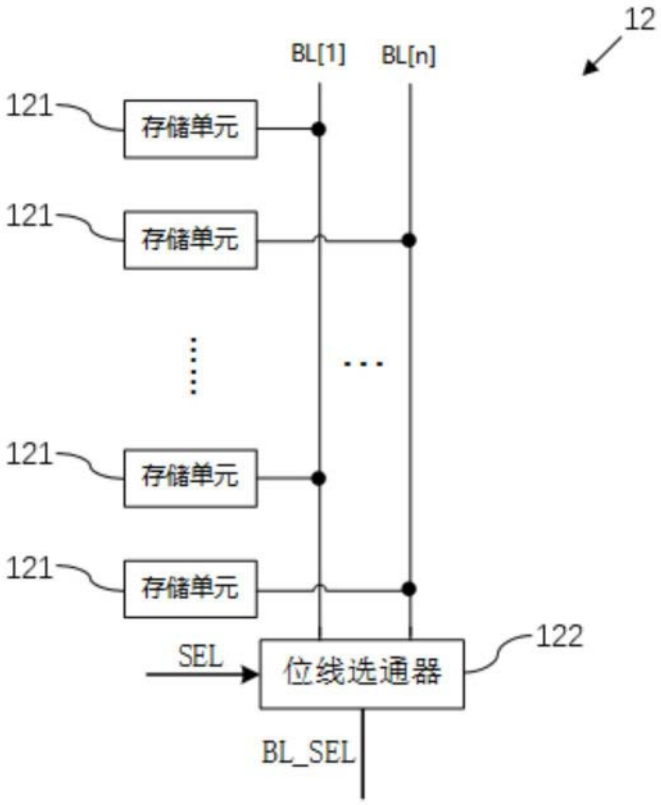

12
BL[1]
BL[n]
121
存储单元
121
存储单元
121
存储单元
121
存储单元
SEL
122
位线选通器
BL_SEL

图2


在第一个时钟周期，向译码阵列输入多个乘数片段，译码阵列向存储器输出相应的译码结果
S1-1
存储器基于译码结果，查找出被乘数的多个部分积，并向加法阵列输出
S1-2
加法阵列基于部分积进行错位相加，得到加法结果，并向移位累加器输出
S1-3
在第二个时钟周期，向译码阵列输入多个乘数片段，译码阵列向存储器输出相应的译码结果
S1-4
存储器基于译码结果，查找出被乘数的多个部分积，并向加法阵列输出
S1-5
加法阵列基于部分积进行错位相加，得到加法结果，并向移位累加器输出
S1-6
移位累加器对两个时钟周期的加法结果进行加权后再进行累加，得到乘法结果
S1-7

图3

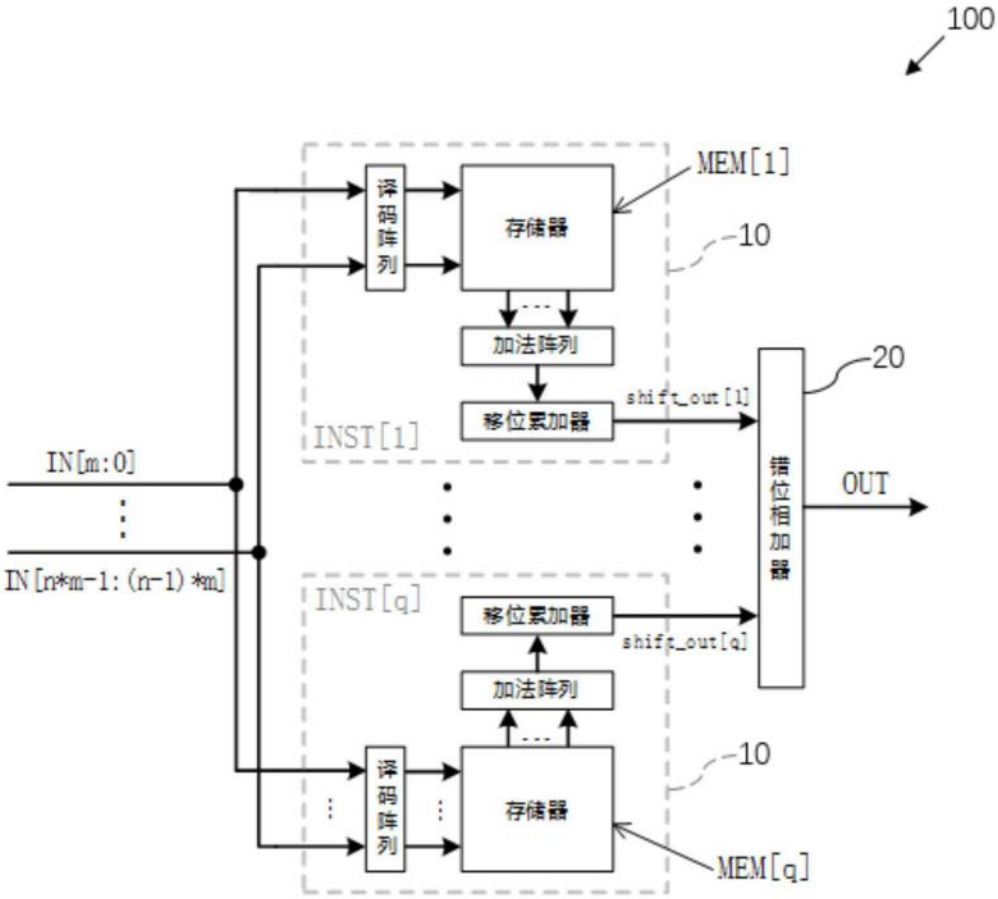

100
MEM[1]
10
译码阵列
存储器
加法阵列
移位累加器
shift_out[1]
INST[1]
20
IN[m:0]
IN[n*m-1:(n-1)*m]
错位相加器
OUT
INST[q]
移位累加器
shift_out[q]
加法阵列
译码阵列
存储器
10
MEM[q]

图4

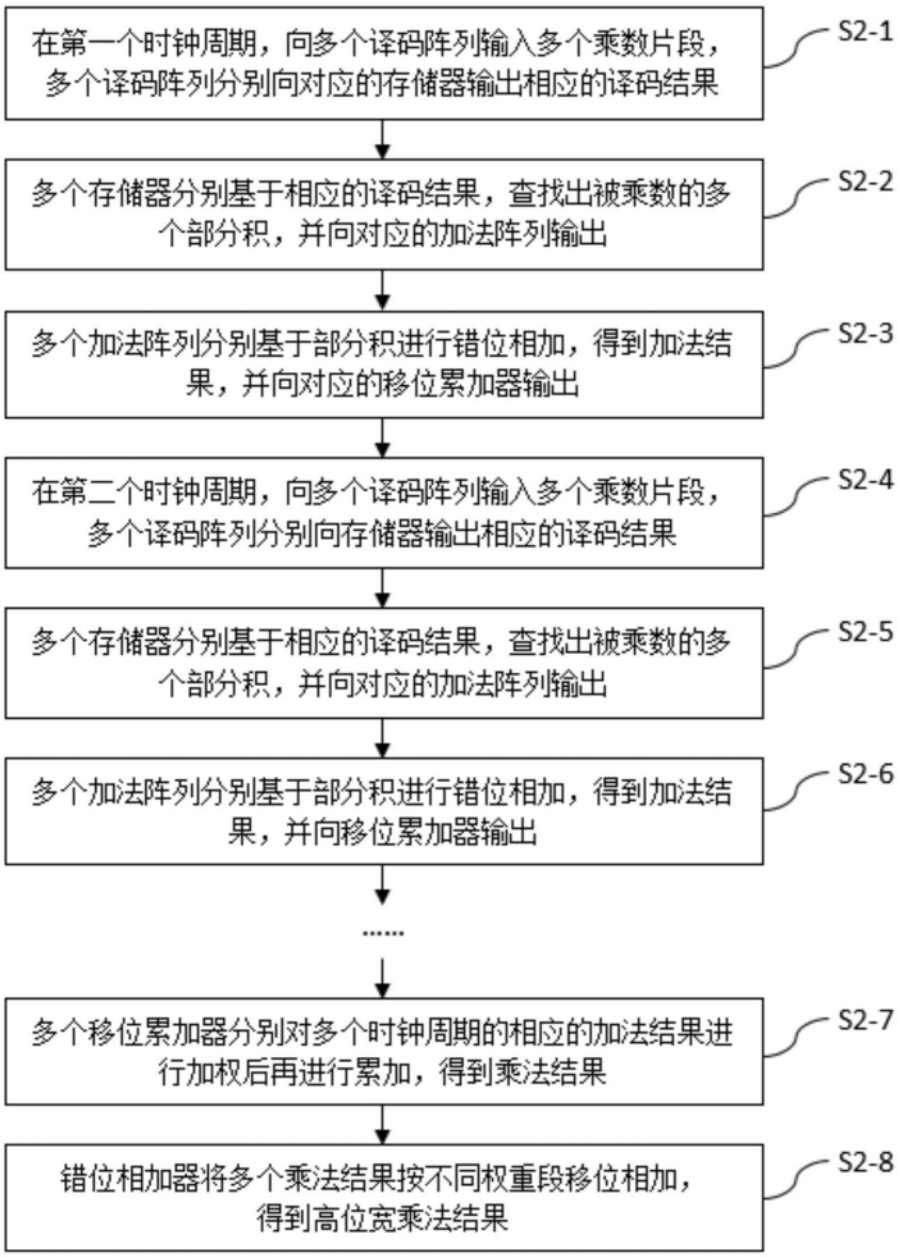

在第一个时钟周期，向多个译码阵列输入多个乘数片段，多个译码阵列分别向对应的存储器输出相应的译码结果
S2-1
多个存储器分别基于相应的译码结果，查找出被乘数的多个部分积，并向对应的加法阵列输出
S2-2
多个加法阵列分别基于部分积进行错位相加，得到加法结果，并向对应的移位累加器输出
S2-3
在第二个时钟周期，向多个译码阵列输入多个乘数片段，多个译码阵列分别向存储器输出相应的译码结果
S2-4
多个存储器分别基于相应的译码结果，查找出被乘数的多个部分积，并向对应的加法阵列输出
S2-5
多个加法阵列分别基于部分积进行错位相加，得到加法结果，并向移位累加器输出
S2-6
.....
多个移位累加器分别对多个时钟周期的相应的加法结果进行加权后再进行累加，得到乘法结果
S2-7
错位相加器将多个乘法结果按不同权重段移位相加，得到高位宽乘法结果
S2-8

图5